CE-ACCE: THE CLOUD ENABLED ADVANCED SCIENCE COMPUTING ENVIRONMENT

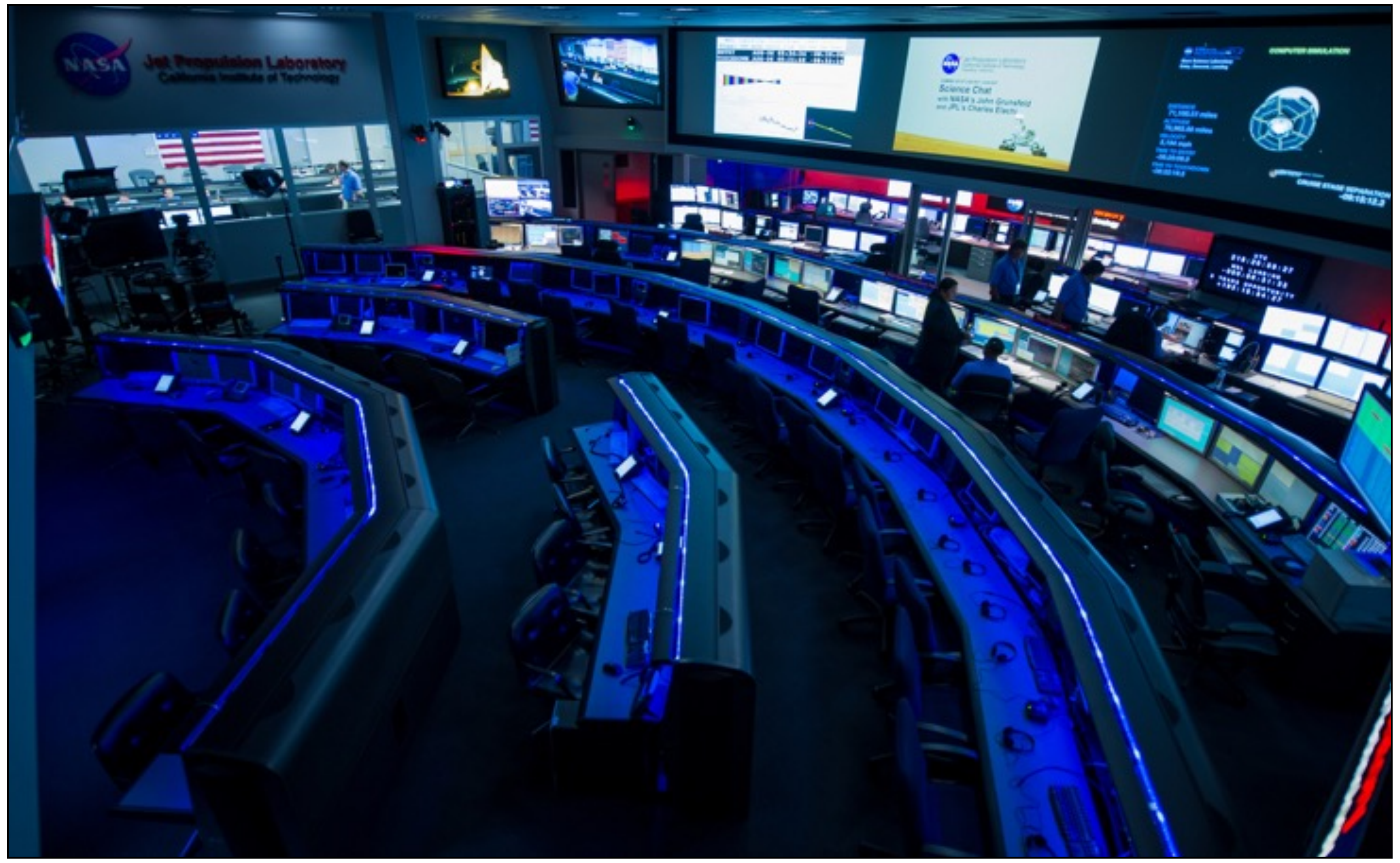
AGU Fall Meeting
December 2017, New Orleans, LA

Luca Cinquini, Dana Freeborn, Sean Hardman and Cynthia Wong [1]
Thanks to: **Benjamin Bornstein, Dan Crichton, Michael Gunson [1]**
[1] California Institute of Technology & NASA Jet Propulsion Laboratory

Funding provided by the Advanced Information System Technologies (AIST) Advanced Concepts program

Outline

- Motivation
- Technologies
- CE-ACCE architecture
- Applications
  - Test Workflow
  - SMAP
  - ECOSTRESS

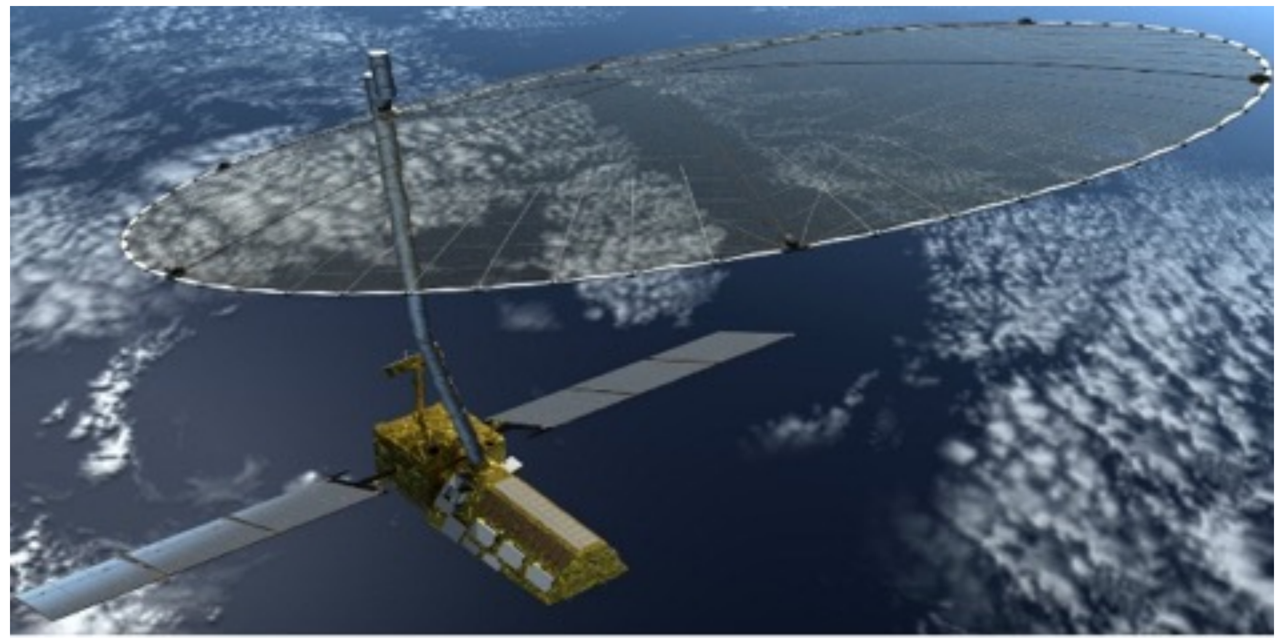


- Lessons Learned
  - The good
  - The bad
- Conclusions
- Future Work

Motivation

- The next generation of NASA observing missions will be collecting and archiving volumes of data which are 1-to-2 order of magnitudes larger than ever before
  - SWOT (Surface Water and Ocean Topography, 2020): 7 TB/day downlinked
  - NISAR (NASA-ISRO Synthetic Aperture Radar, 2020): 85 TB/day downlinked, 140 PB archived over 3 years
  - Comparison: the *total* volume of all NASA data in EOSDIS archive is approx. 22 PB

- In preparation, NASA and JPL have started to design and evaluate new hardware infrastructures and software architectures that are capable of handling these unprecedented data volumes
  - Must be scalable, re-usable, resilient, and cost-effective



NISAR

ECOSTRESS

SWOT

Motivation (cont.)

- Recently, the accessibility of the Cloud, coupled with new advances in containerization technologies (e.g. Docker) and orchestration (e.g. Docker Swarm, Kubernetes, AWS) has enabled a new emerging paradigm:
  - Define standard, reusable system architectures
  - Plug in mission specific configuration: workflows and PGEs ("Product Generation Executables")
  - Seamlessly deploy to internal cluster, commercial Cloud, or hybrid system

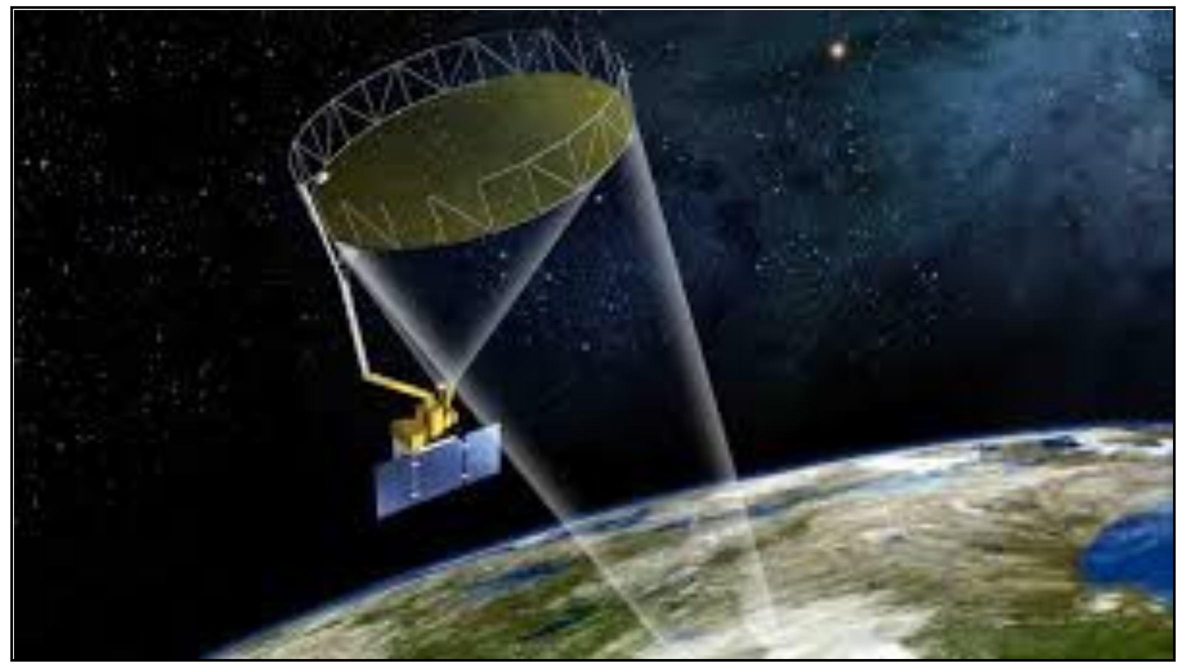
- This talk will demonstrate the validity of this approach for ACCE, a data processing framework that has been developed at JPL for over a decade, and successfully applied to several airborne and satellite missions

ACCE

ACCE ("Advanced sCience Computing Environment")
is a re-usable SDS ("Science Data System") environment for small missions

- Historically funded by JPL to provide a packaged SDS for cost-capped missions
- Objectives: reduce risk, lower cost, and provide ubiquitous access to data
- ACCE software package is based on Apache OODT for data processing, cataloging and access
- ACCE has been repeatedly demonstrated in a production environment for remote sensing mission (both airborne and satellite), achieving TRL-9 recognition

CARVE

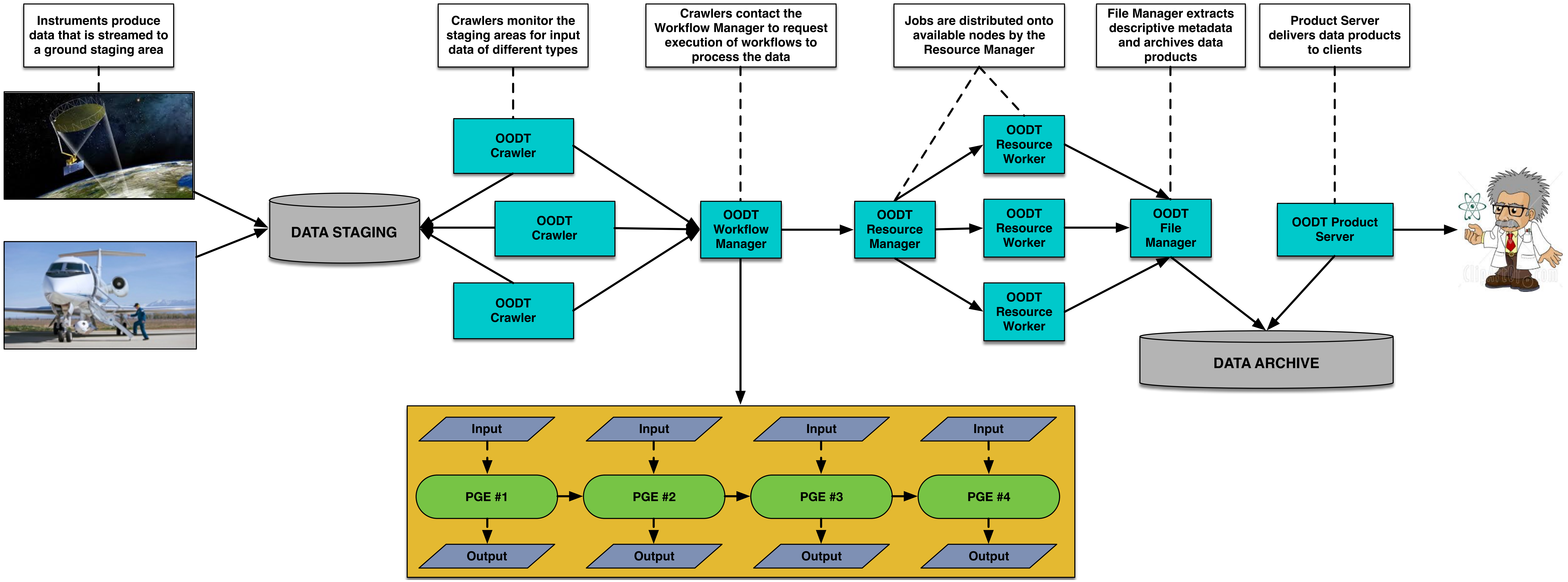


SMAP

OMG

Apache OODT

OODT ("Object Oriented Data Technology"): open source framework for managing scientific data through its full lifecycle, including data processing, indexing, archiving and access

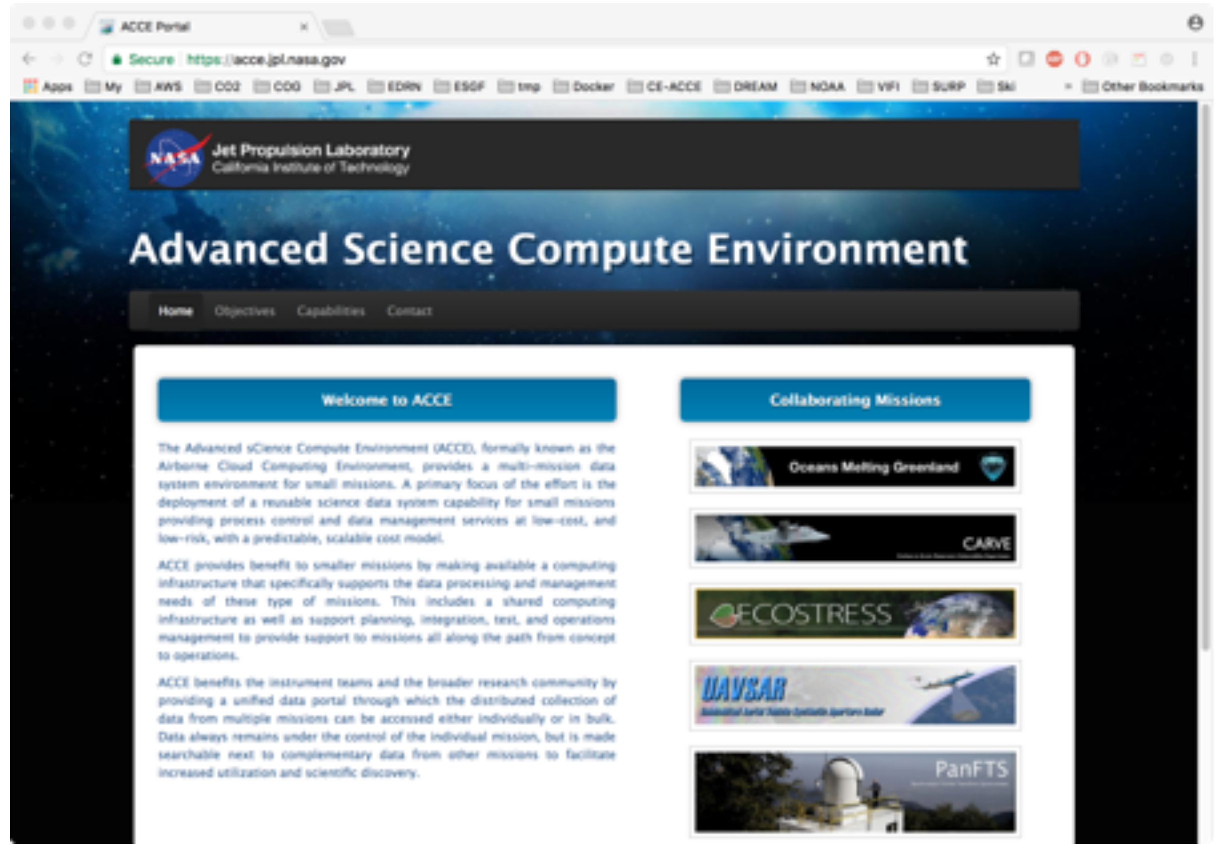


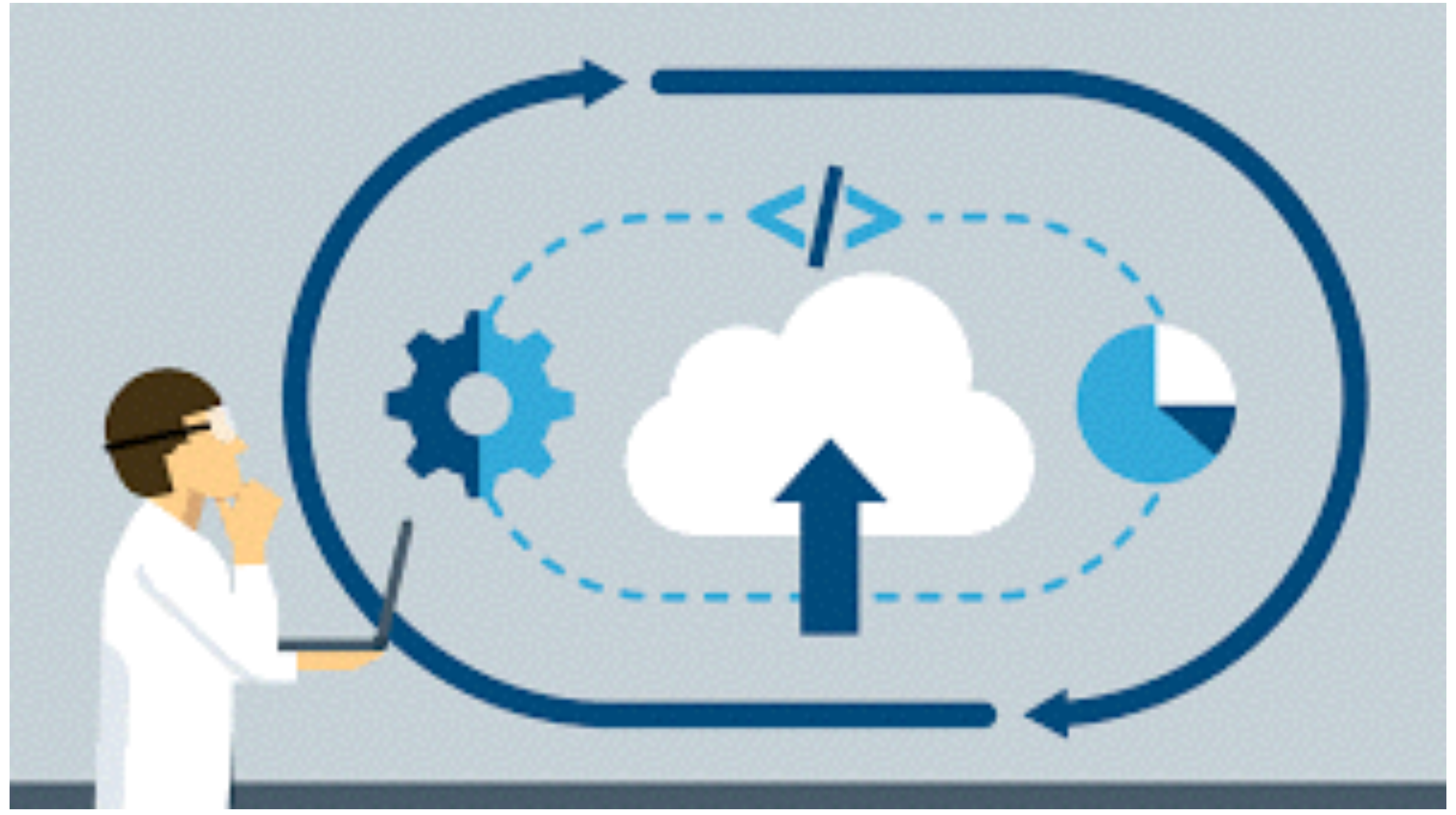
Example OODT architecture to process data from a NASA mission

The Cloud-Enabling of ACCE

- In FY17, JPL funded an internal task to provide an approach for running ACCE on the Cloud:
  - to enable massive scaling and bursting
  - to build on the open source legacy of OODT and proven reliability of ACCE
  - to enable deployment on multiple Cloud environments including private and commercial Clouds

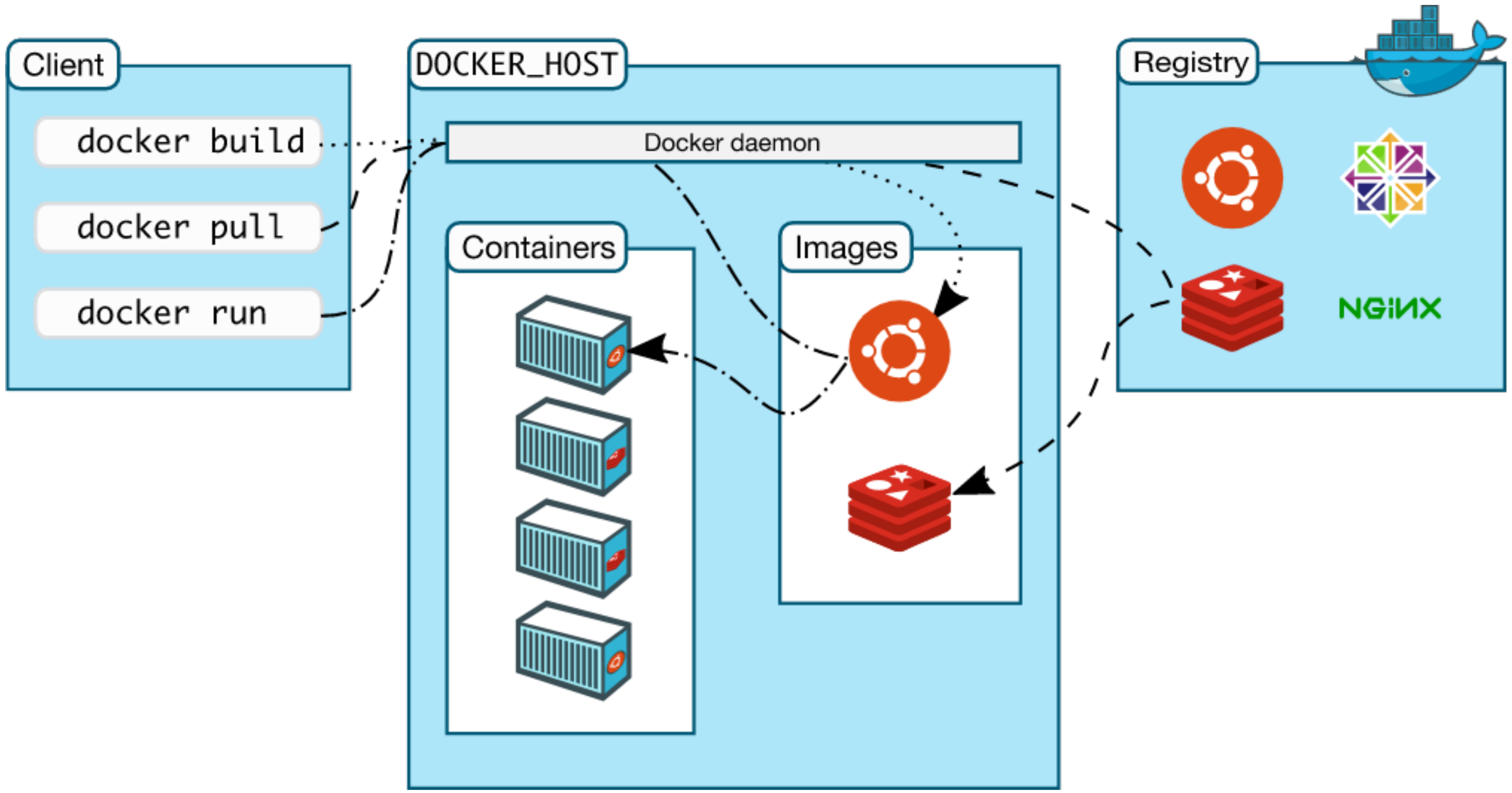
- To accomplish this task, the team designed and implemented a new ACCE architecture based on Docker ("ACCE/Docker"), where individual OODT services run as independent yet interacting software containers



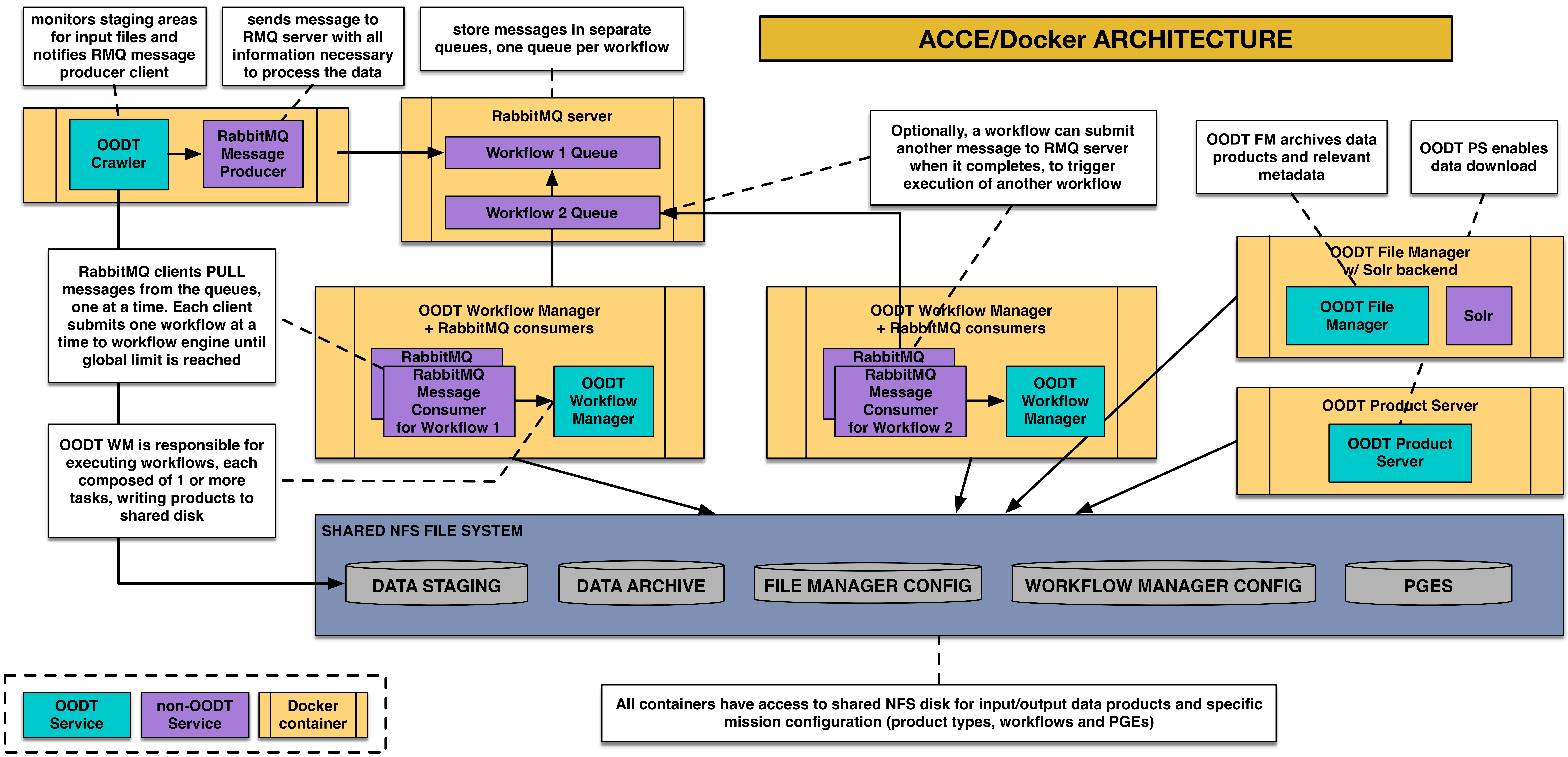


Docker Overview

- Docker is the industry leading containerization technology - "<u>build, ship & run</u>":
  - applications are built as software images that include the application itself, all required dependencies, and "just enough OS" to run them
  - images are uploaded to common repositories
  - images are deployed as "black boxes" on any platform running a Docker engine



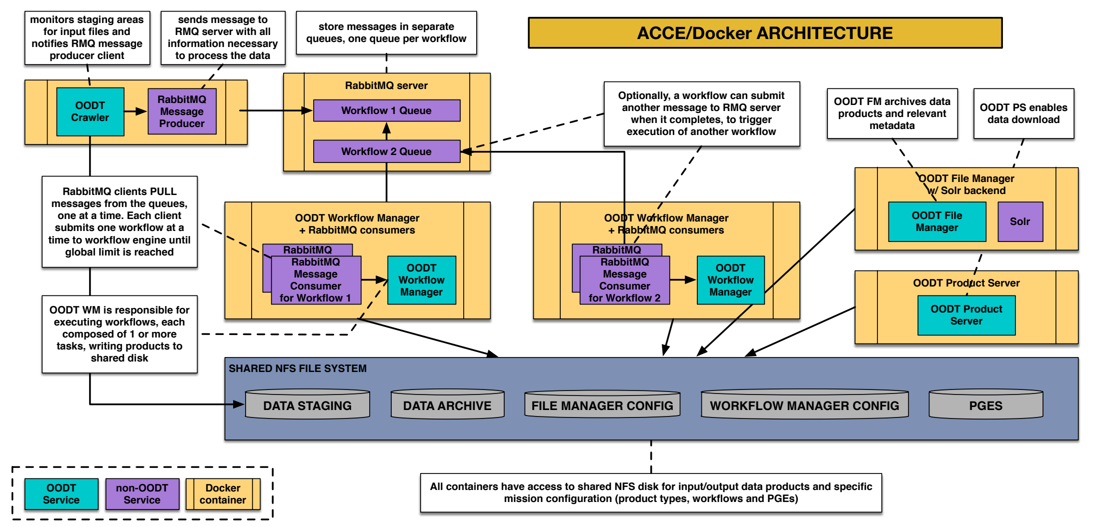
ACCE/Docker Reference Architecture



CE-ACCE architecture advantages: easy to deploy, portable, pluggable, scalable

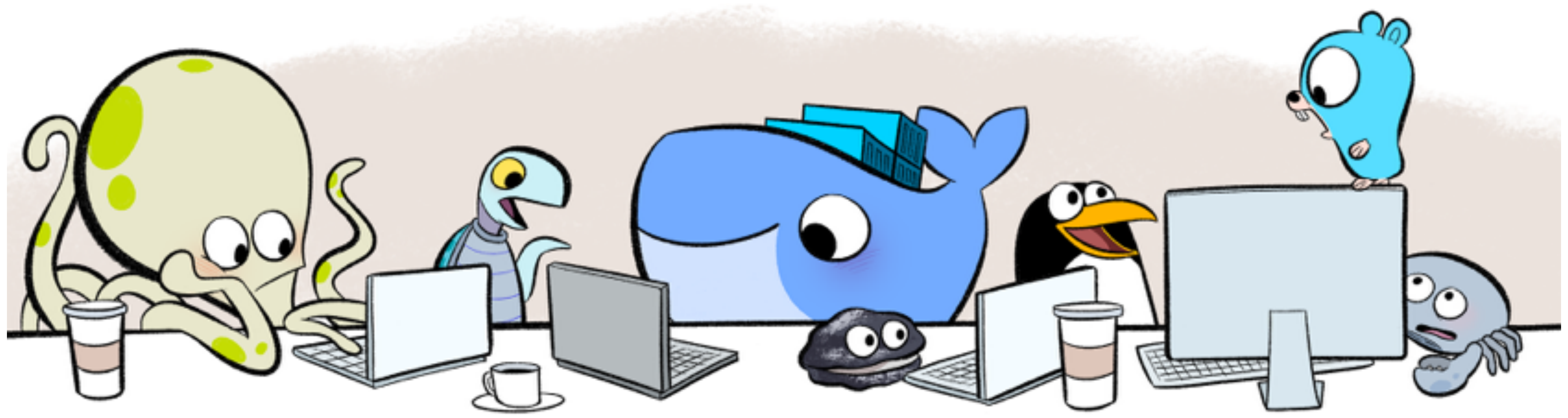
The Cloud-Enabling of ACCE

- Each OODT service is encapsulated as a Docker container
- OODT services (FM, WM) are setup to read mission specific configuration (workflows, PGEs, data types) from pre-defined location
- Input/output data mounted on a shared disk partition
- New RabbitMQ message broker enables a more decoupled and scalable processing environment
- Docker containers can interact because of networking provided by the orchestration engine



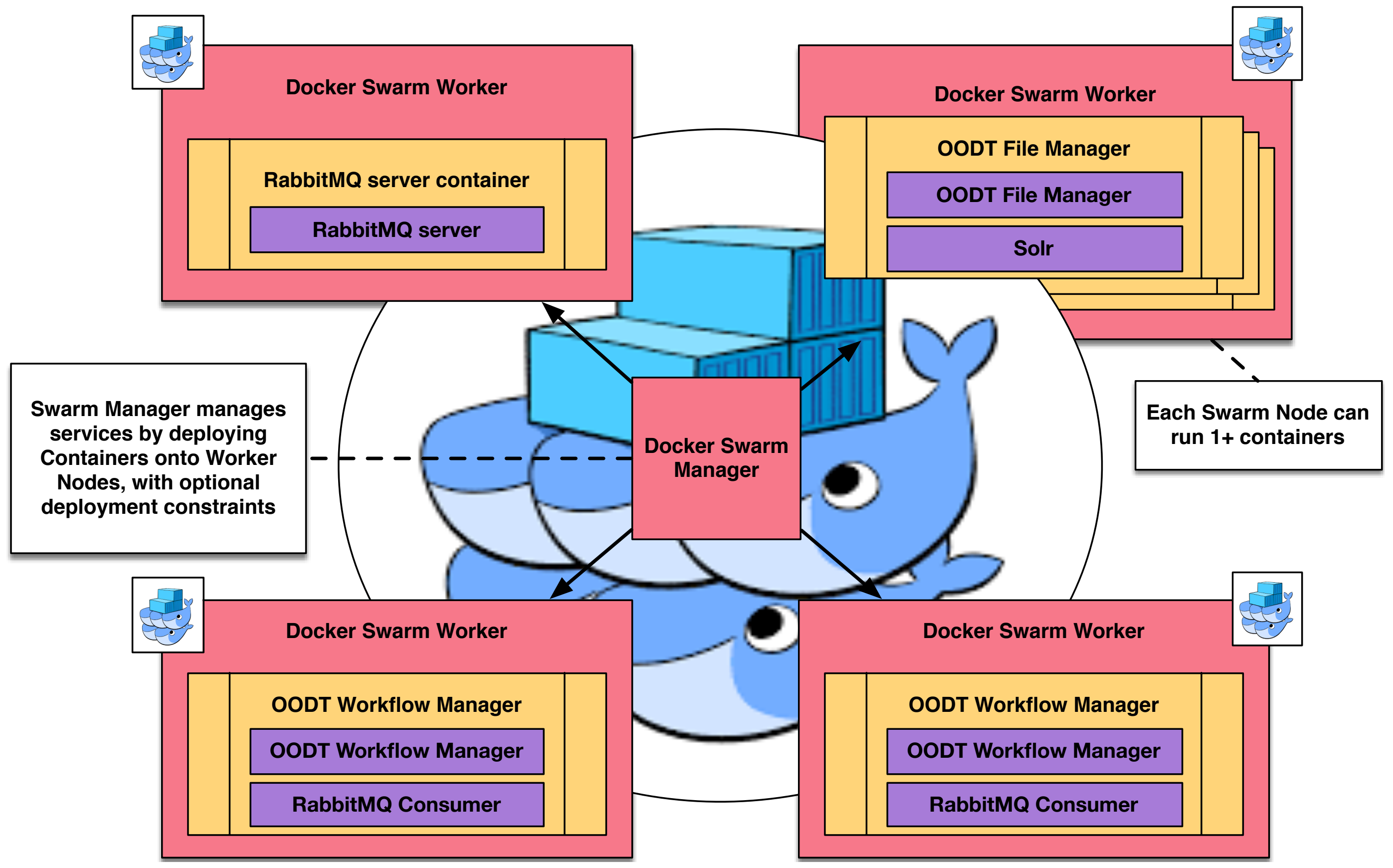
Advantages of ACCE/Docker Architecture

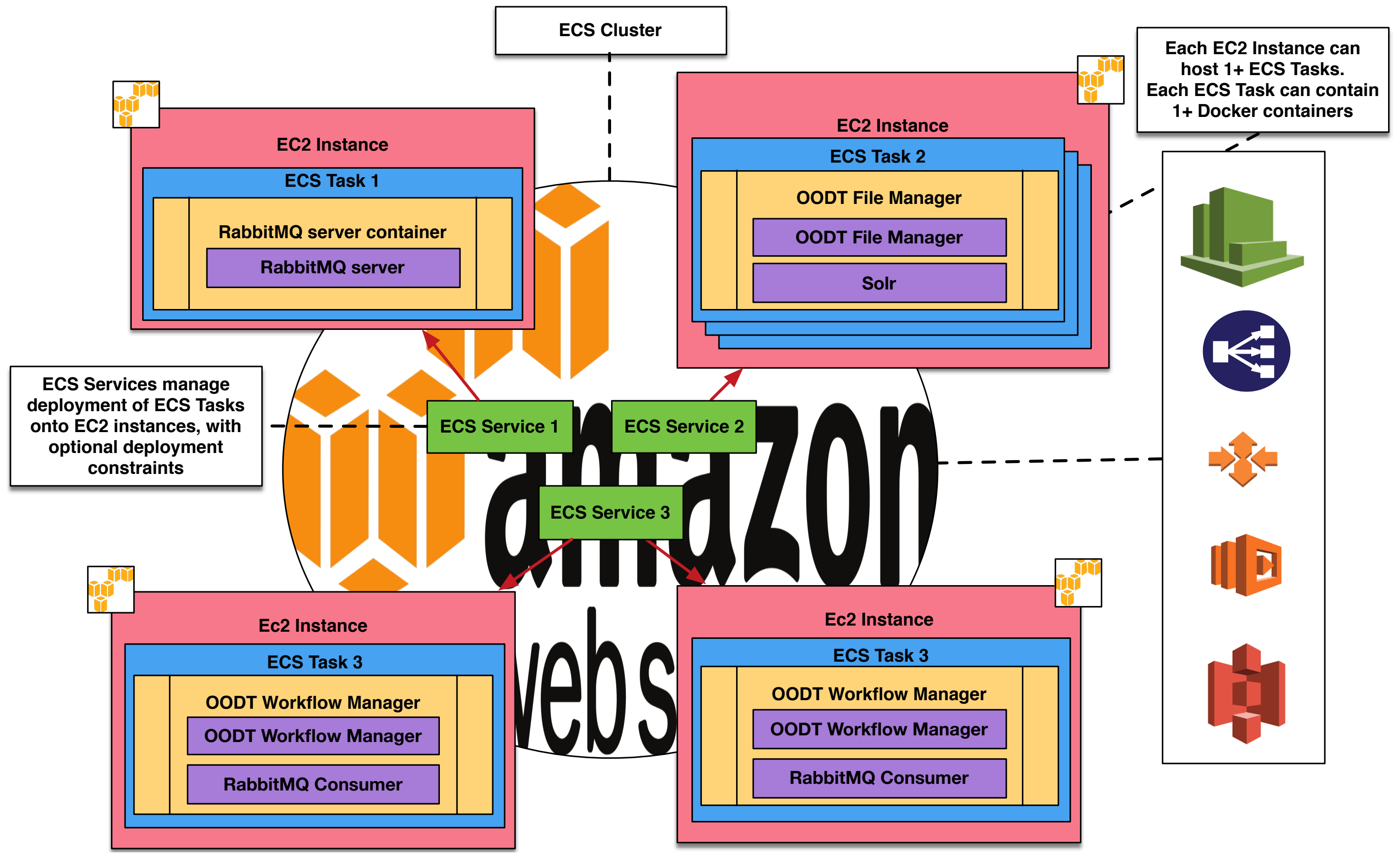
- Easy to deploy: just download images and start containers, no compilation needed
- Portable: can run on any platform where Docker is installed (developer's laptop, internal JPL cluster, private and commercial Clouds)
- Pluggable: missions re-use the same services, provide their own workflow configuration and executables
- Scalable: number of Docker containers can be scaled using standard orchestration tools such as Docker Swarm, Kubernetes, AWS EC2 Container Services, etc…



ACCE Deployment on Amazon Cloud

- We deployed the ACCE/Docker architecture on the Amazon Cloud and applied to both current and future data processing use cases
- We experimented with 2 orchestration engines: Docker Swarm and Amazon ECS





Docker Swarm

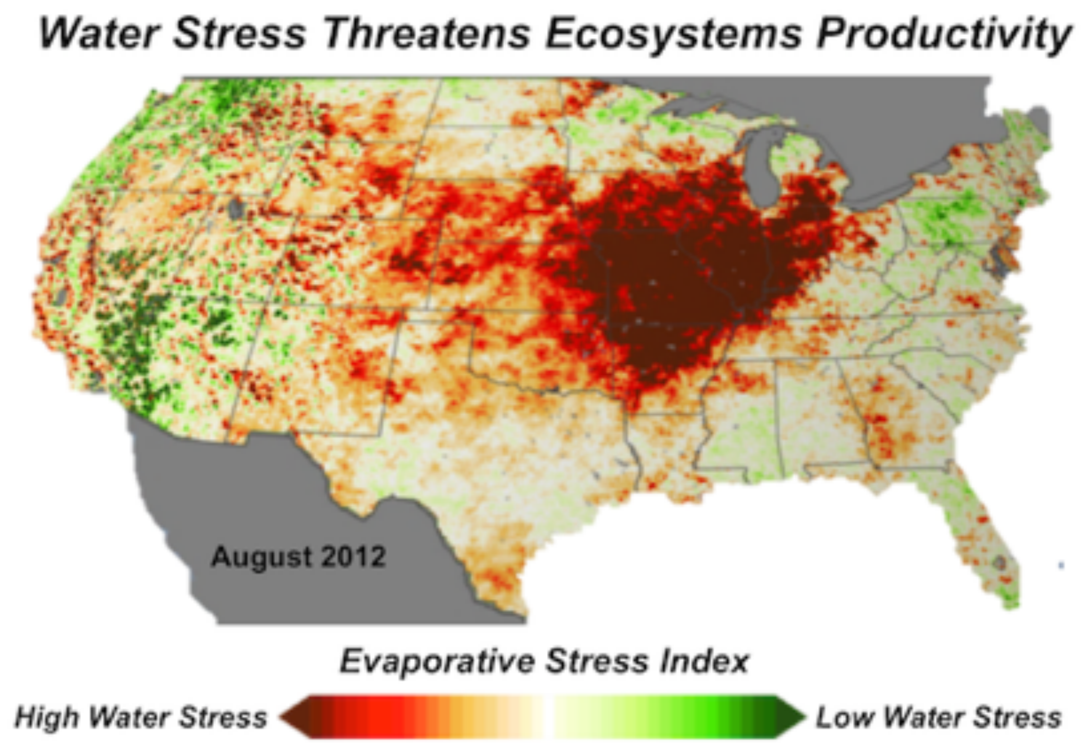
- Intrinsic Docker orchestration engine
- Available wherever Docker is deployed
- Provides easy scaling, load balancing, automatic failover recovery, high availability through routing mesh, …
- ..but no auto-scaling

Amazon ECS (“EC2 Container Service”)

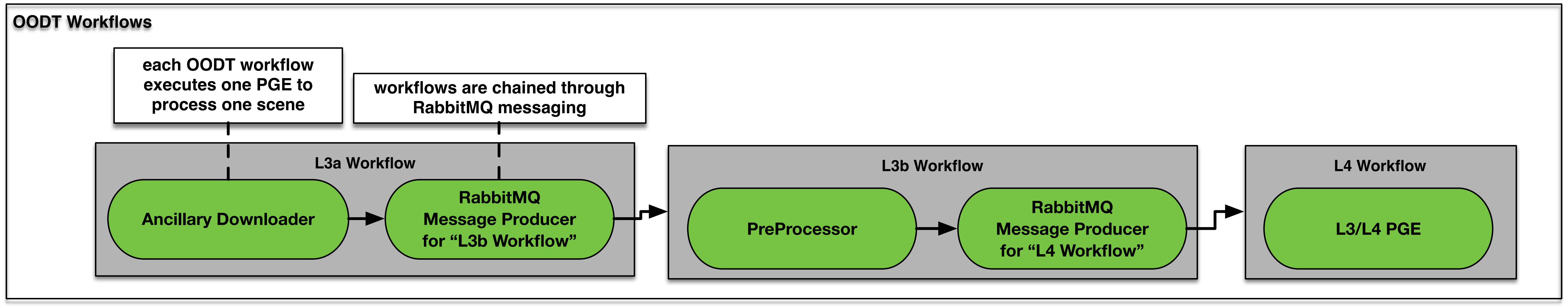
- AWS environment for running Docker apps
- Clusters of EC2 instances with Docker installed
- ECS Services run N instances of ECS Tasks
- ECS Tasks include 1+ Docker containers
- Integrates w/ AWS S3, Lambda, Auto-Scaling, CloudWatch, ELBs, …
- …but tied to AWS specific framework and APIs

Application to ECOSTRESS L3/L4 Data Processing

- As a first example, we applied the ACCE/Docker architecture to simulate and benchmark the expected L3/L4 data processing pipeline for the ECOSTRESS mission
- ECOSTRESS ("ECOsystem Spaceborne Thermal Radiometer Experiment on Space Station"): upcoming NASA mission that will fly a thermal radiometer aboard the ISS to study changes in vegetation due to climate change and water availability

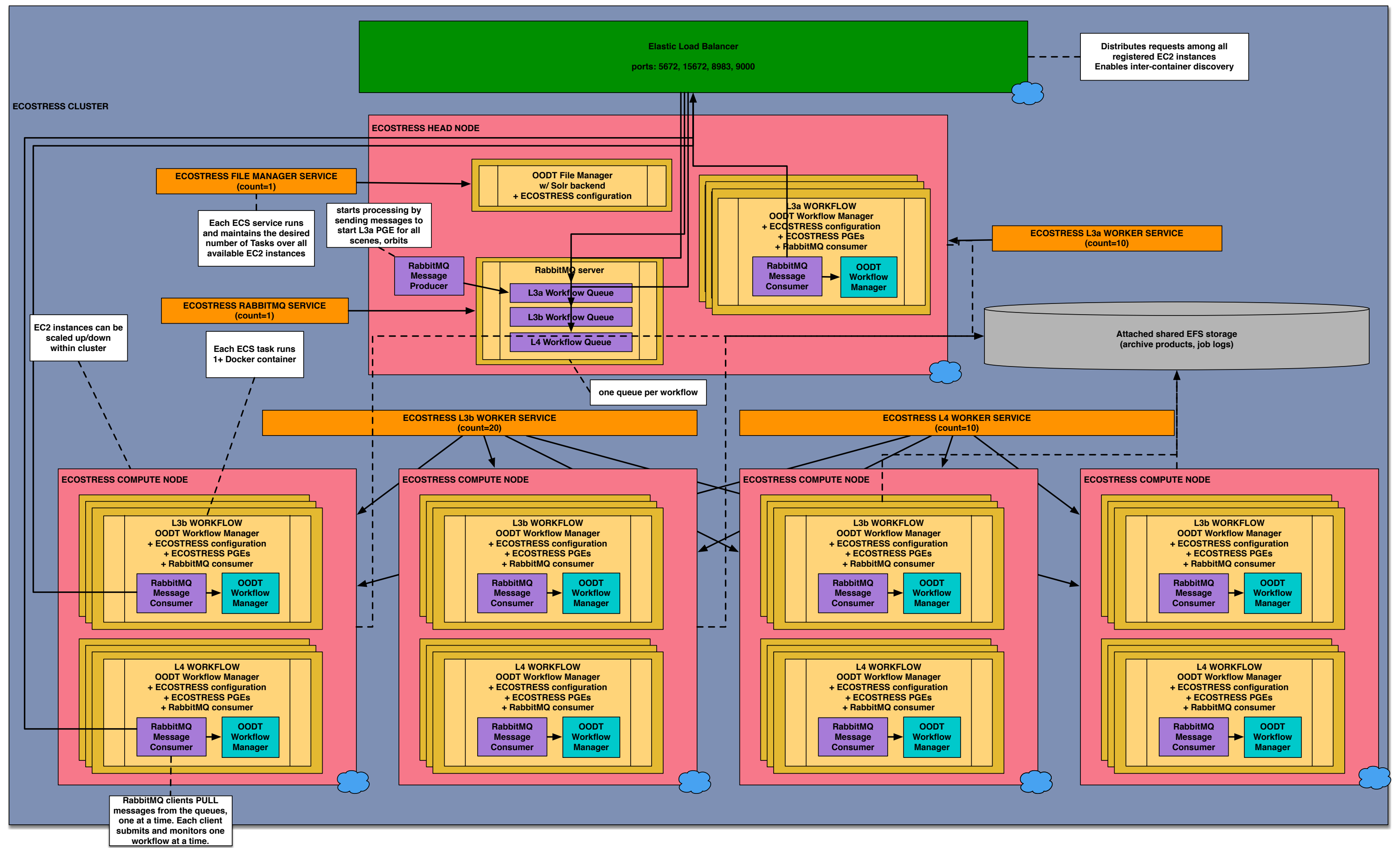


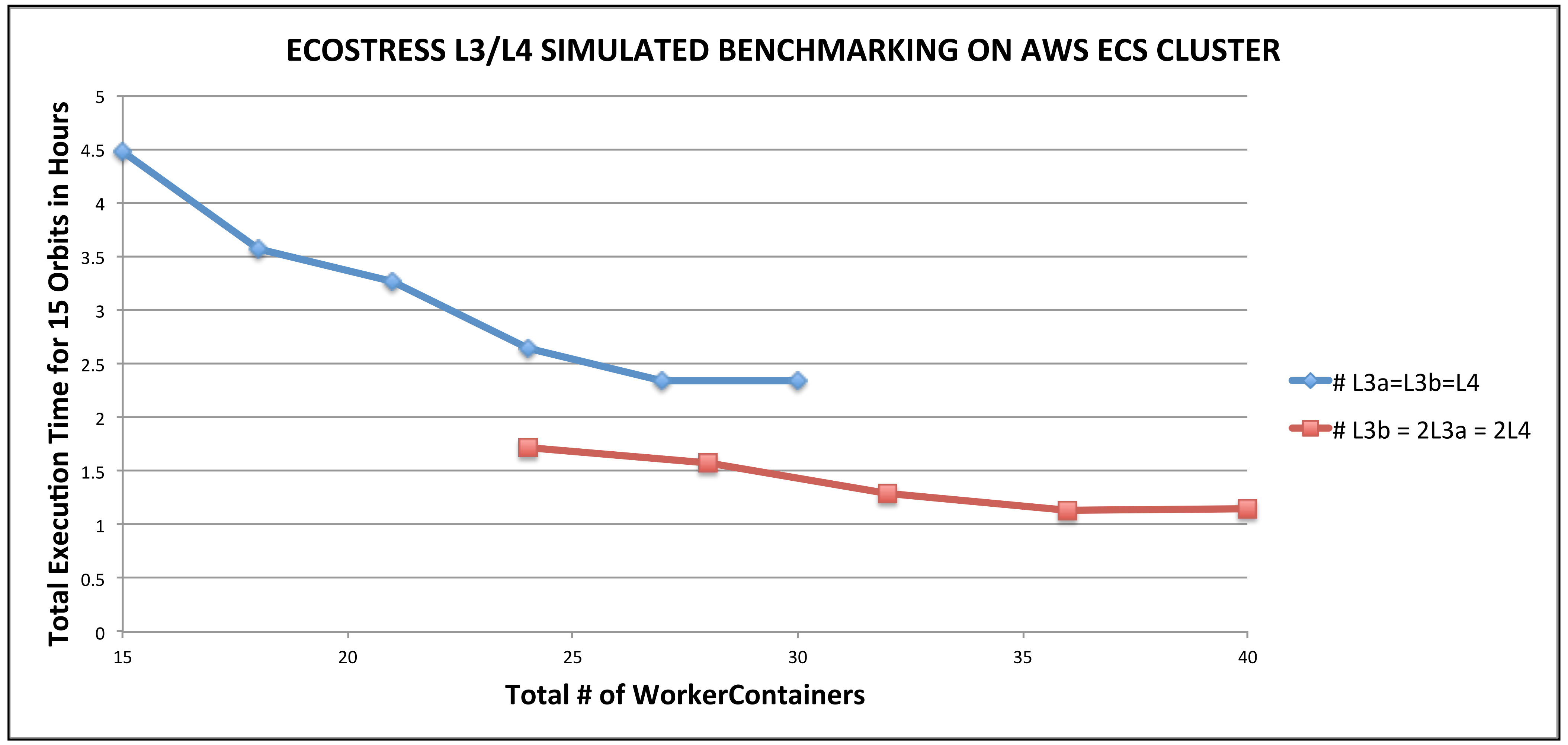
- Expected L3/L4 processing: each image or scene needs to be processed by a sequence of 3 PGEs running on 2 separate nodes ("head node" and "compute" node)
- We setup a sequence of 3 OODT workflows each running a single PGE, where simulated compute times for each PGE were based on the most recent L3/L4 PGE benchmark tests



Application to ECOSTRESS L3/L4 Data Processing

- We deployed ACCE/Docker on AWS ECS cluster composed of 1 head + 4 compute nodes
- We benchmarked the processing of a full day of data (15 ISS orbits = 142 scenes) as a function of the number of OODT WM containers instantiated on each node

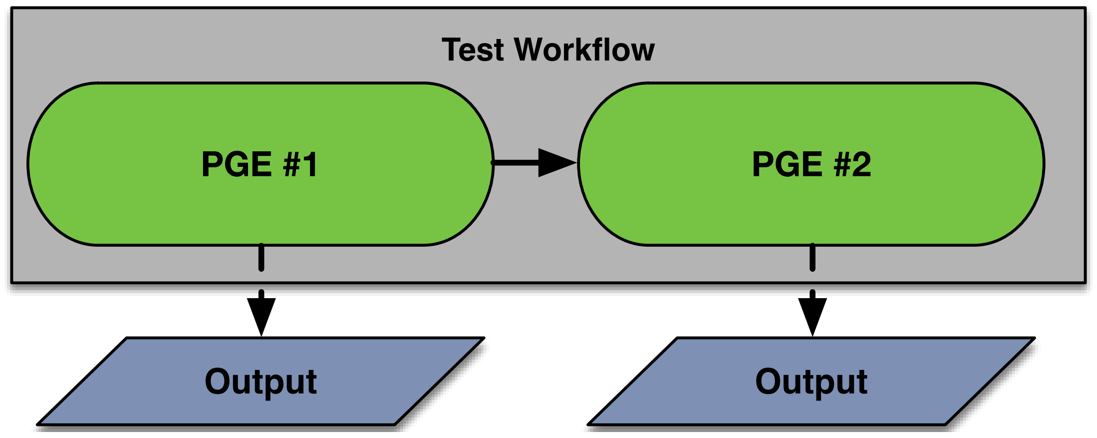


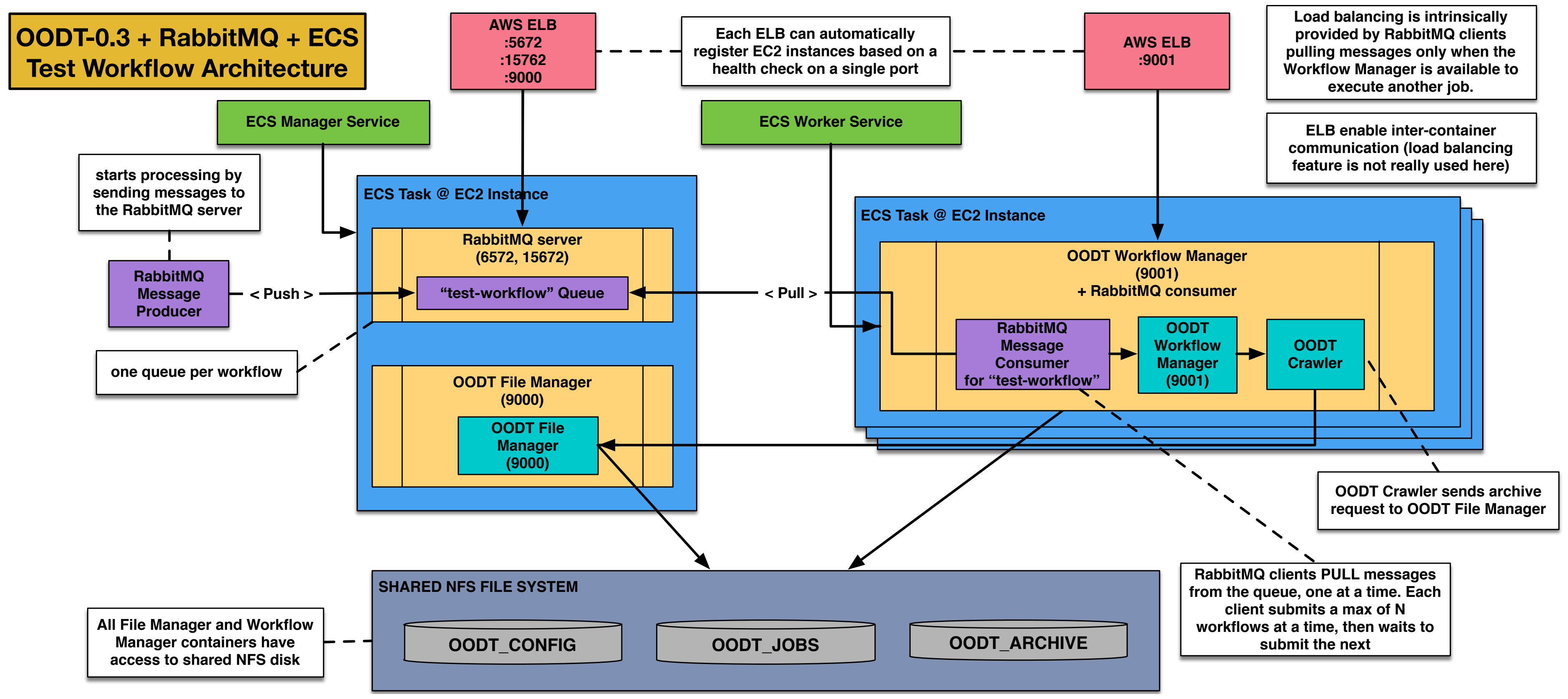


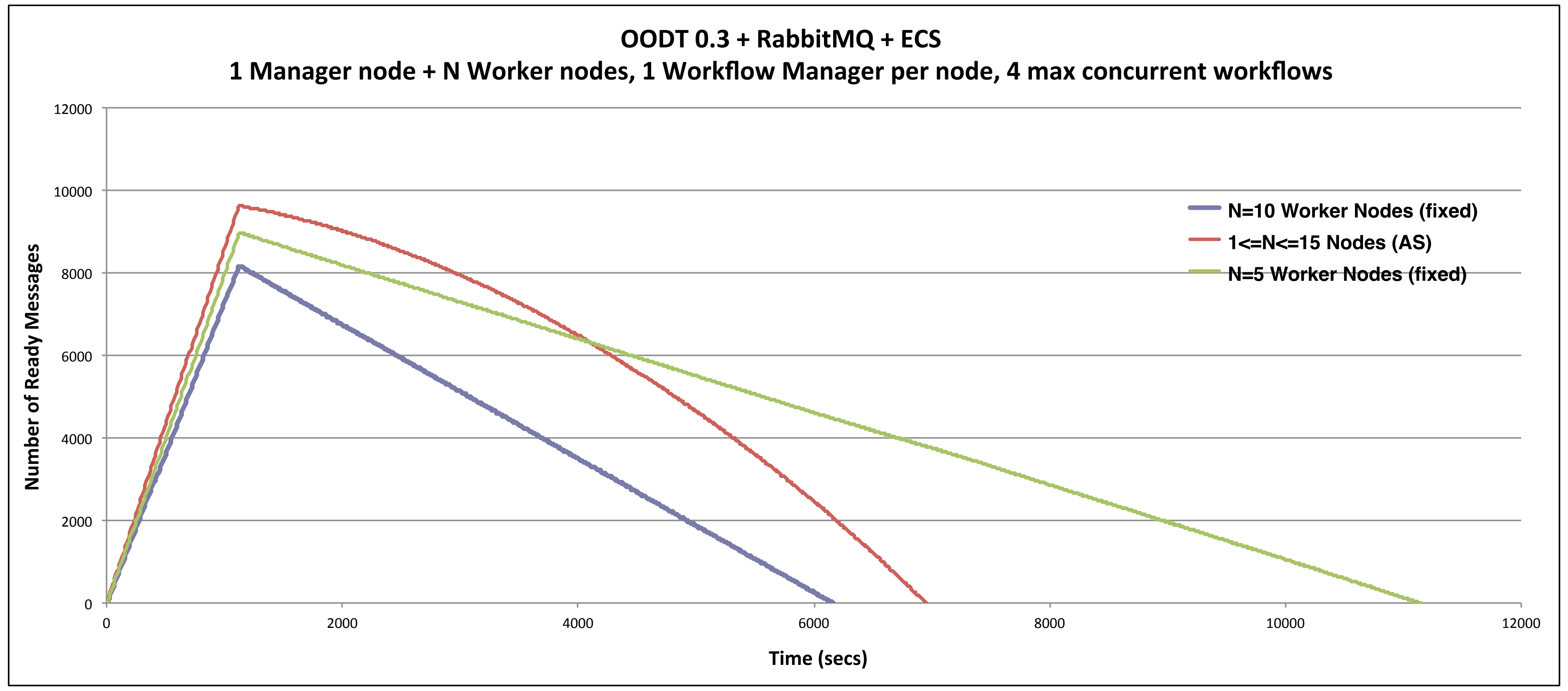
- Conclusions:
  - it was straightforward to deploy and configure ACCE/Docker to execute the ECOSTRESS L3/L4 data processing
  - the architecture scaled well by increasing the number of worker containers
  - a full day of L3/L4 data could be processed in about an hour on a medium size EC2 cluster

Scalability Studies on AWS ECS

- We assessed the scalability of the ACCE/Docker framework using Amazon ECS:
  - Test workflow composed of 2 PGEs, each writing out an output file
  - 1 EC2 Manager Node hosting FM and RMQ server containers
  - N EC2 Worker Nodes hosting WM containers (where N can be either fixed, or auto-scaled)
  - Executing 10,000 workflows = 20,000 PGEs



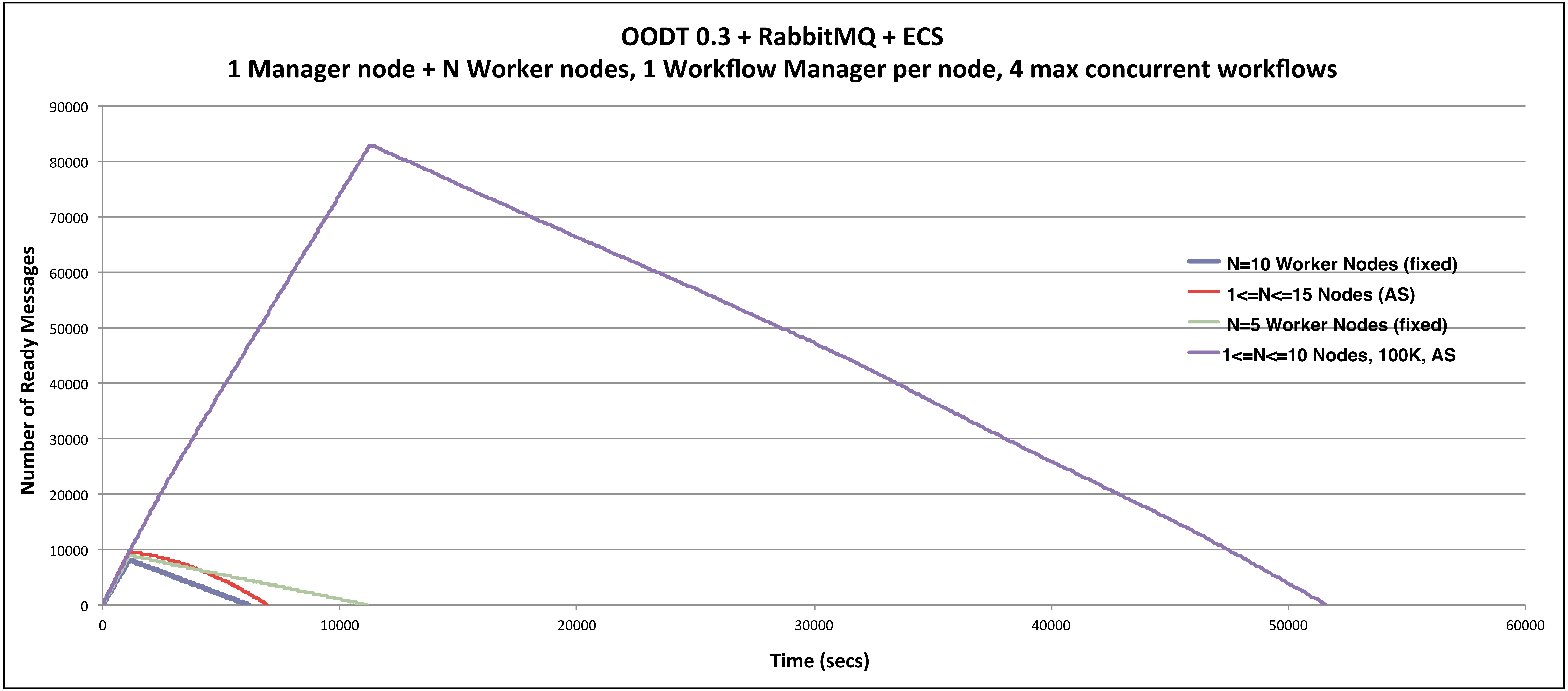




- Results:
  - All workflows completed
  - ACCE/Docker framework can sustain rates of several tens of thousands of workflows/ PGEs per day as expected from upcoming NASA missions SWOT and NISAR

More Scalability Studies on AWS ECS

- Pushing the limit: 100,000 workflows (i.e. 200,000 PGEs) in about 14 hours…

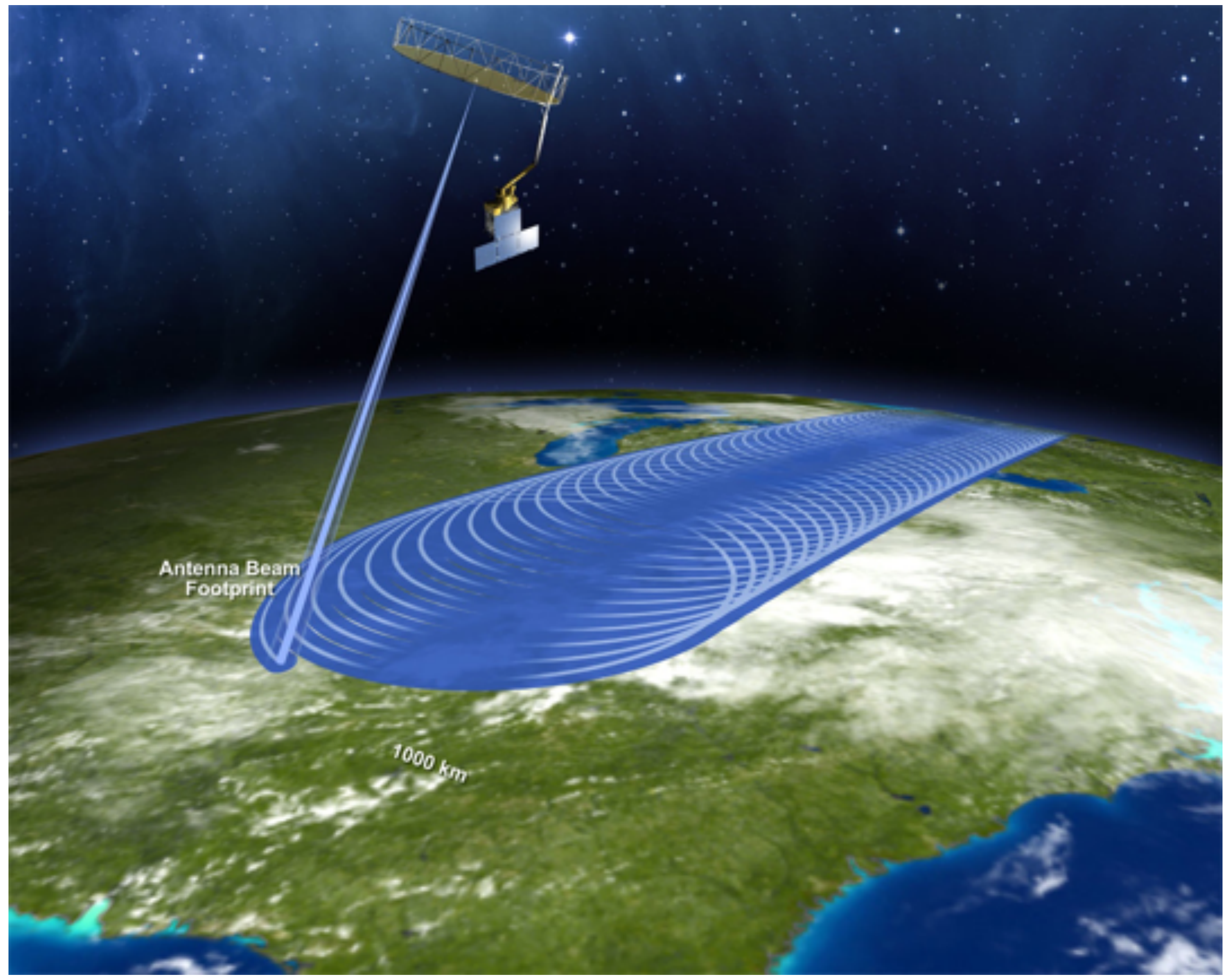


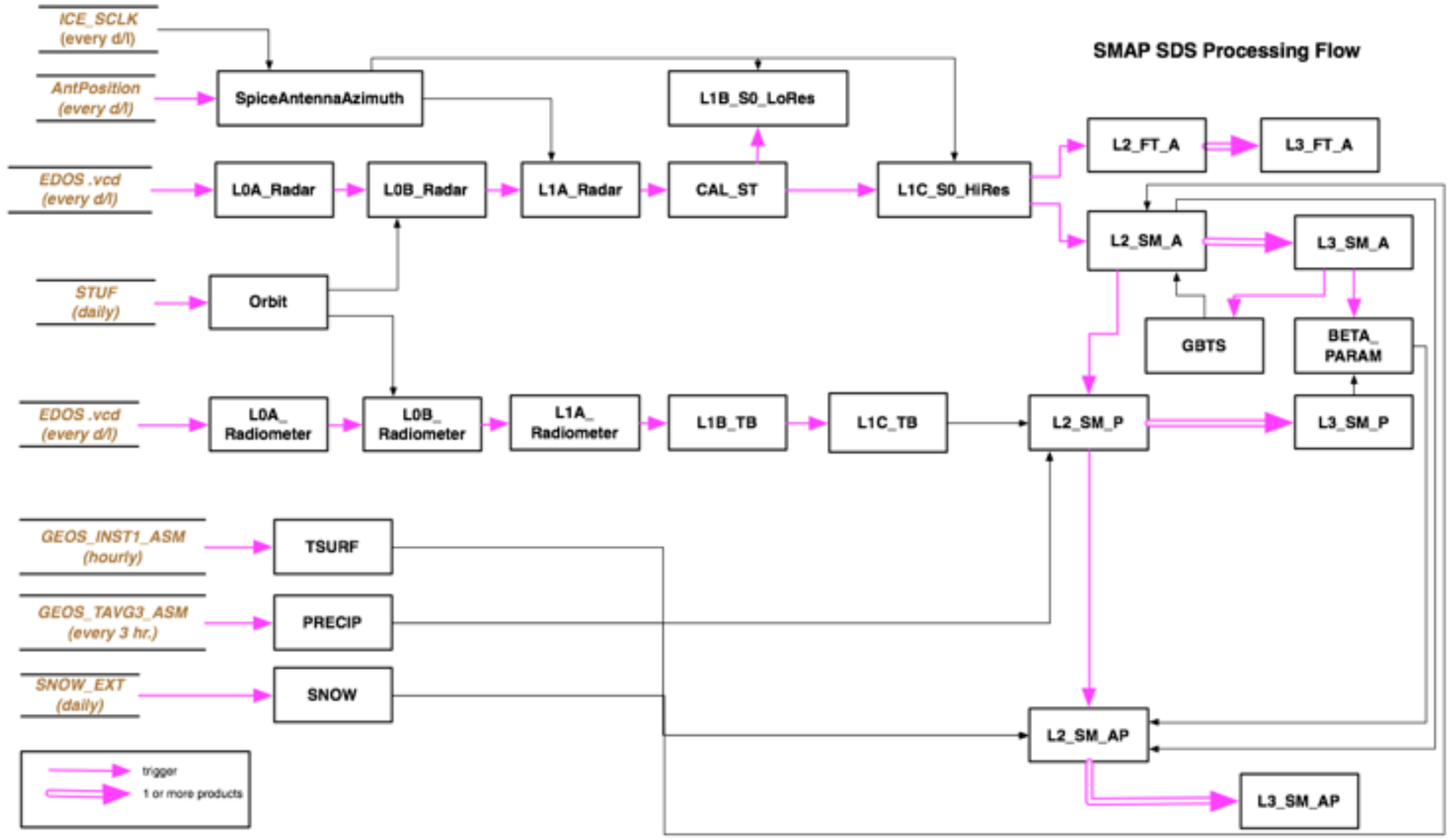
Application to SMAP Test Suite

- We experimented with deploying and running the SMAP SDS as Docker containers in different environments: JPL internal cluster and Amazon self-provisioned EC2s

- SMAP ("Soil Moisture Active Passive"): orbiting NASA satellite that measures the amount of water in the top 5cm of soil everywhere on the Earth surface

- The existing SMAP SDS includes 13 crawlers, tens of inter-dependent PGEs



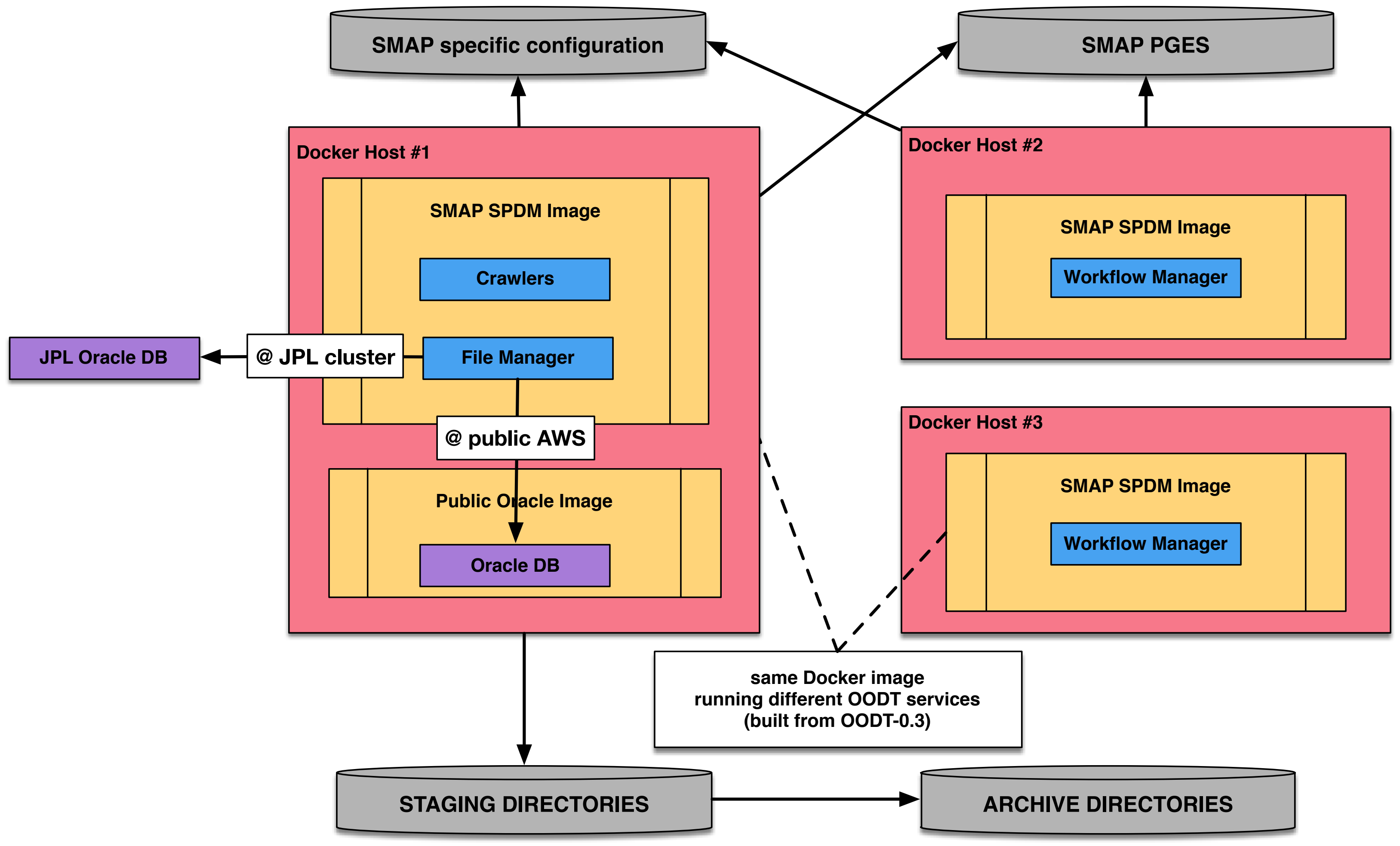


Application to SMAP Test Suite

- Because SMAP already has a legacy working SDS, we did not break up the data services and extract the SMAP specific configuration to be used by the standard ACCE/Docker framework: rather, we created a Docker image hosting the complete SMAP SDS

- The same SMAP SDS image was used to start different OODT services (FM+crawlers or WM) on different hosts, depending on startup parameters

- Deployed on a Docker Swarm composed of 1 manager node and 2 worker nodes



- Results:
  - We were able to successfully execute the full SMAP SDS regression suite on JPL internal cluster and Amazon Cloud, using 1 container per node
  - Test suite got stuck when using more than 1 container per host
  - Task funding ran out before the problem could be debugged…

Application to ECOSTRESS L1/L2 Data Processing

- We also deployed the ECOSTRESS L1/L2 data processing on the Cloud, using existing actual PGEs

- We proceeded in a similar manner to the SMAP use case:
  - One single Docker image containing ECOSTRESS SDS, starting different OODT services (FM+crawlers or WM) depending on configuration
  - Deployed on JPL internal cluster
  - Used Docker Swarm for orchestration and networking

- Accomplishments:successfully executed L0B & L1A RAW PGEs

- Issues found during execution of PGEs:
  - Dependencies on system libraries, VICAR libraries, and ECOSTRESS Python modules for PGEs
  - Higher version of OS required
  - Access to PGEs' supporting directories such as Camera model, MERRA, or SPICE, etc.
  - Docker images size > 6.6GB

- Conclusion: the existing ECOSTRESS L1/L2 SDS can be run on the Cloud, but not using a standard, re-usable micro-services architecture without modifications to the PGE deployment

Conclusions and Lesson Learned

- Recent advances in system technologies (Cloud, Docker, orchestration engines) are enabling a new paradigm for designing science data systems that are easier to deploy, reusable, and more scalable

- One such SDS is the newly re-architected ACCE/Docker framework, which we have demonstrated to be suitable for deploying and scaling data processing of small NASA missions on the Cloud (up to 100K workflows/day)

- Two equally challenging issues arise when porting a legacy SDS to run in the Cloud:
  - The data management package -which was tested for a specific hardware/network/storage configuration- might not work in a high availability, dynamic environment, and might need to be partially re-architected
  - Existing PGEs may not be portable, i.e. they might depend on a specific OS version and system libraries - in our case, this was the biggest challenge when adapting the existing SMAP and ECOSTRESS systems

- Consequently, new SDS systems must be architected for the Cloud from the very beginning:
  - Build portable PGEs, optimally as Docker containers
  - Test deployment in a dynamic environment, with a variable number of nodes
  - Plan for hardware/network failures, build resiliency into the system

Current and Future Work

- Application of ACCE/Docker to AMIGHO
  - Processing on the Cloud of hydrological data collected by GNSS (Global Navigation Satellite System) enabled stations around the U.S.
- Define and use a “template PGE”
  - “blue-print” for developing PGEs that can be run by the ACCE/Docker framework, with different methods invoked at specific stages of the workflow lifecycle
- Port the ACCE/Docker architecture to Kubernetes and OpenShift
  - Cloud-agnostic orchestration engine to enable seamless deployment on AWS, GCE, Azure, etc.
- Instrument RabbitMQ broker to enable automatic recovery from Cloud failures
- Develop standard UI for monitoring and book keeping, package as Docker container